



Univerza v Mariboru

Fakulteta za organizacijske vede

Diplomsko delo visokošolskega strokovnega študija
Organizacija in management informacijskih sistemov

SPLETNA STRAN ZA IZMENJAVO ŠTUDIJSKIH GRADIV

Mentor: doc. dr. Borut Werber

Kandidat: Marko Bitenc

Kranj, december 2013

ZAHVALA

Zahvaljujem se mentorju doc. dr. Borutu Werberju za nasvete in pomoč pri izdelavi diplomske naloge.

Zahvaljujem se tudi mag. Alenki Tratnik, ki je lektorirala mojo diplomsko nalogo.

POVZETEK

Diplomska naloga predstavlja izgradnjo dinamične spletne strani, ki bo delovala na osnovi programskega jezika PHP. Spletna stran je namenjena izmenjavi študijskih gradiv med študenti Fakultete za organizacijske vede v Kranju in omogoča bodočim študentom pomoč pri študiju.

Spletna stran omogoča prikaz vsebine glede na izbiro uporabnika. V ta namen smo uporabili bazo podatkov, ki nam omogoča izpis izbranih podatkov. Uporabnik lahko vanjo tudi sam dodaja predmete in zapiske.

Uvodni del naloge opisuje teoretične osnove statične in dinamične spletne strani z namenom ugotoviti razlike osnovnih pojmov med njima, vključuje pa tudi opis teoretičnih osnov spletnih jezikov in uporabljenih orodij.

V nadaljevanju naloge sta opisana postopek izgradnje dinamične spletne strani in uporaba orodij.

Zaključek naloge vsebuje ugotovitve ter možnosti nadaljnjega razvoja spletne strani.

KLJUČNE BESEDE:

- spletna stran
- PHP
- HTML
- CSS

ABSTRACT

The diploma thesis presents the construction of a dynamic website that will function on the basis of a programming language PHP. The website is intended for the exchange of study materials for the students of the Faculty of Organizational Sciences in Kranj and aids future students in their studies.

The website displays the chosen content according to the user's choice. To this end, we used a database that enables the display of the chosen data. The user can also add single objects and records in the database.

The introductory part of the thesis describes the theoretical basis of static and dynamic web pages aiming at identification of differences between the two basic concepts, in addition to theoretical description of the basics of Web languages and tools used.

Furthermore, the description of the process of building dynamic website and the use of tools is given.

In conclusion to this work, the findings and possibilities for the website's further development are presented.

KEYWORDS:

- website
- PHP
- HTML
- CSS

KAZALO

1. UVOD.....	1
1.1. Predstavitev problema.....	1
1.2. Namen naloge.....	1
1.3. Cilji naloge.....	1
1.4. Metode dela.....	2
2. TEORETIČNE OSNOVE.....	4
2.1. Statična spletna stran.....	4
2.2. Dinamična spletna stran.....	5
2.3. Spletna stran za izmenjavo dokumentov.....	6
2.4. Optimizacija spletne strani.....	7
3. SPLNETNI JEZIKI IN ORODJA.....	9
3.1. HTML.....	9
3.2. PHP.....	10
3.3. CSS.....	11
3.4. mySQL.....	12
3.5. phpMyAdmin.....	13
3.6. Xampp.....	14
4. STRUKTURA SPLETNE STRANI.....	15
4.1. Naslovna stran.....	15
4.1.1. Predloga naslovne strani.....	16
4.1.2. Grafično oblikovanje.....	17
4.2. Registracija.....	24
4.2.1. Grafična predloga.....	24
4.2.2. Izdelava registracije.....	25
4.3. Iskalnik zapiskov.....	28
4.3.1. Opis vmesnika za izbiro zapiskov.....	28
4.3.2. Sistem delovanja vmesnika za izbiro zapiskov.....	29
4.3.3. Izdelava podatkovne baze.....	30
4.4. Prikaz zapiska.....	31
4.4.1. Sistem delovanja.....	32
4.5. Dodajanje zapiskov in predmetov.....	34
4.5.1. Sistem delovanja.....	35
5. OPTIMIZACIJA SPLETNE STRANI.....	37
6. ANALIZA SWOT.....	39
7. ZAKLJUČEK.....	41
8. UGOTOVITVE.....	41
9. MOŽNOSTI NADALJNJEGA RAZVOJA.....	42
LITERATURA IN VIRI.....	43

1. UVOD

1.1. PREDSTAVITEV PROBLEMA

Ker smo v času študija imeli na razpolago zelo malo študijskih zapiskov, smo se domislili, da bi bilo koristno narediti spletno stran, kjer bi na enem mestu bili vsi zapiski Fakultete za organizacijske vede v Kranju in na razpolago vsakemu študentu te fakultete.

V ta namen smo se odločili izdelati dinamično spletno stran v PHP jeziku za dinamičen prikaz vsebine; le-ta bi prikazovala zapiske, ki bi jih uporabnik izbral, obenem pa bi tudi sam lahko dodajal zapiske v bazo podatkov.

Zaradi želje po še večjem znanju smo se odločili, da sami naredimo spletno stran brez uporabe že obstoječih CMS-jev.

1.2. NAMEN NALOGE

Namen diplomske naloge je izdelati spletno stran in tako omogočiti študentom, da lažje sledijo nek predmet, ga bolje razumejo ter posledično dobijo boljšo oceno na izpitu.

Študentje nimamo spletne strani, kjer bi lahko na enem mestu izmenjevali zapiske, zato predvidevamo, da bo izdelava takšne spletne strani študentom v času študija v veliko pomoč, saj bo na tem mestu vsak registrirani uporabnik imel vse zapiske na dosegu roke.

1.3. CILJI NALOGE

Cilji naloge so:

- izdelati grafično zasnovo spletne strani,
- izdelati relacijsko bazo podatkov,
- izdelati kodo v PHP za pridobivanje zapiskov,
- izdelati kodo v PHP za dodajanje predmetov in zapiskov,

- izdelava registracije uporabnika,
- izdelava prijave uporabnika in
- izdelava zaščite spletne strani.

Za grafično predlogo spletne strani smo se odločili za svetle tone barv in enostaven videz zaobljenih linij. Ker je spletna stran dinamična, je zato uporaba relacijske baze podatkov obvezna, saj se z uporabnikovo interakcijo vsebina spreminja.

Glavni cilj diplomske naloge je izdelati programsko kodo v PHP, kjer bo uporabnik lahko samostojno izbiral med naslednjimi kriteriji: program, smer, letnik in predmet.

Prav tako smo se odločili izdelati programsko kodo v PHP za dodajanje predmetov in zapiskov, tako da bo lahko vsak uporabnik imel možnost dodajati svoja študijska gradiva.

Zaradi večjega nadzora in varnosti nad uporabniki ter podatki smo se odločili še za izdelavo registracije in prijave uporabnikov, ter se tako v večjem delu zaščitili pred zlorabo in verodostojnostjo zapiskov, česar v celoti ni mogoče izvedljivo.

1.4. METODE DELA

Osnovna metoda dela vsebuje uporabo teoretičnega in praktičnega dela. Praktični del postavitve spletne strani ne vsebuje že narejenega CMS sistema. Za izdelavo dinamične spletne strani smo uporabili prototipni pristop. Prednost prototipne metode je v tem, da lahko uporabnik lažje ugotovi, ali aplikacija ustreza vsem njegovim zahtevam. S prototipom lahko uporabnik eksperimentira, ga oceni ali dopolni, dokler dovolj dobro ne ustreza zahtevam uporabnika. Vir: (<http://www.swe.uni-linz.ac.at/publications/pdf/tr-se-94.02.pdf>, 26. 11. 2013).

Najprej smo preučili obstoječe stanje in analizirali najboljše rešitve, nato smo načrtovali strukturo in oblikovne predloge. Sledila je tehnična izvedba, kjer smo naredili dva

prototipa, in na koncu testirali prototipne rešitve. Ta metoda zajema samostojno ustvarjanje in oblikovanje osnovnih elementov izgradnje spletne strani.

Kot je navedeno (Zupančič, 2007), je prototipni pristop primeren kadar:

- Zahteve uporabnikov so nejasne in/ali “mehke”.
- Zahteve se s časom (hitro) spreminjajo.
- Inovativna aplikacija in/ali tehnologija.
- Aplikacija bo pomembno vplivala na delo organizacije.
- Razmeroma majhne dimenzije projekta oz. dela projekta.
- Razmeroma majhno število uporabnikov.
- Razmeroma kratek čas razvoja projekta.
- Kadar želimo doseči oz. je nujno tesno sodelovanje uporabnikov.

Za samo postavitev strani smo uporabili označevalni jezik HTML (HyperText Markup Language).

Za videz smo uporabili CSS (Cascading Style Sheet), ki služi prikazu HTML elementov na spletni strani, s tem pa se izognemo odvečnemu ponavljanju kode.

Z orodjem PhpMyAdmin smo ustvarili relacijsko bazo podatkov ter uporabili programski jezik PHP za celovito funkcionalnost spletne strani.

Za delovanje spletne strani smo uporabili orodje Xampp, ki služi kot lokalni strežnik.

2. TEORETIČNE OSNOVE

2.1. STATIČNA SPLETNA STRAN

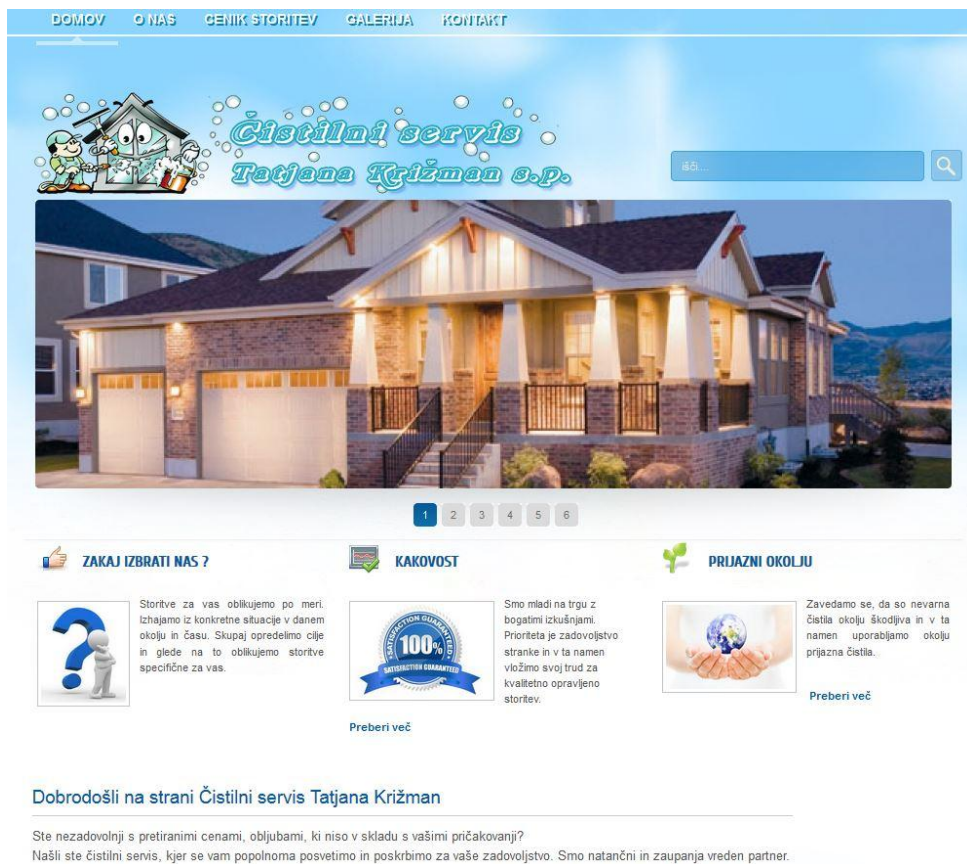
Statična spletna stran je namenjena predstavitvi dejavnosti ali storitvi. pri čemer nam ni treba spreminjati vsebine na spletni strani, saj se tekst in slike ne menjajo. Vsi obiskovalci statične spletne strani vidijo isto vsebino. Spremembe vsebine so možne samo z interakcijo avtorja, ki poseže v kodo. Za tehniko izdelave je dovolj HTML v povezavi s CSS.

Prednost statične spletne strani je v tem, da ima statičen URL in jo tako spletni iskalniki lažje najdejo, saj se ji naslov ne spreminja.

Statična spletna stran je enostavna za izdelavo in zato cenovno ugodna.

Vir: (<http://izdelava-spletnih-strani-1a.blogspot.com/2010/06/staticne-ali-dinamicne-spletne-strani.html>).

Slika 1 prikazuje primer statične spletne strani, kjer se vsebina ne menja dinamično.



Slika 1: Primer statične spletne strani

Vir: (<http://www.cisto-je.si> 15. 8. 2013)

2.2. DINAMIČNA SPLETNA STRAN

Vsebina dinamične spletne strani se z interakcijo uporabnika spreminja. Primerna je za spletne trgovine, portale in spletne strani, kjer se vsebina menja. V večini se dinamične spletne strani izdeluje s CMS-ji, saj omogočajo hitrejšo postavitev.

Dinamične spletne strani so zahtevne za izdelavo in zahtevajo obširno znanje spletnih orodij, kot so baze podatkov, ter spletnih jezikov, kot so: PHP, .ASP, AJAX, JQuery in Javascript.

Vir: (<http://izdelava-spletnih-strani-1a.blogspot.com/2010/06/staticne-ali-dinamicne-spletne-strani.html>, 20. 6. 2013).

Slika 2 prikazuje spletno trgovino mimovrste.com, ki je tipičen primer dinamične spletne strani.

Slika 2: Primer dinamične spletne strani

Vir: (www.mimovrste.com, 15. 9. 2013)

2.3. SPLETNA STRAN ZA IZMENJAVO DOKUMENTOV

Spletna stran za izmenjavo dokumentov je dinamična, saj se vsebina spreminja z interakcijo uporabnika. Uporabili smo programski jezik PHP, ki poskrbi, da se določen del kode zažene in da dobimo prikazane želene podatke, ki se nahajajo v podatkovni zbirki.

Uporabnik z interakcijo elementov pošlje zahtevek za določen dokument, ta se izvede v obliki SQL stavka, ki pošlje zahtevo na strežnik. Strežnik nato posreduje izbrano vsebino iz baze podatkov in jo prikaže uporabniku. V obratni smeri lahko želeno vsebino tudi

dodajamo v bazo; v tem primeru potrebujemo formo za vnos podatkov, na podlagi katere z SQL stavkom povemo, kaj in kam naj se podatki shranijo v podatkovno zbirko na strežnik.

2.4. OPTIMIZACIJA SPLETNE STRANI

Zaradi množice poplav informacij, prisotnih na internetu, je zelo pomembna optimizacija spletne strani, saj le tako izboljšamo našo vidljivost na spletu in s tem pripeljemo uporabnike na našo spletno stran. Ko uporabnik v iskalnik vnese določeno zaporedje znakov ali ključno besedo, to pomeni, da išče informacijo o točno določenem izdelku, storitvi ali produktu. To pa pomeni, da so iskalci z določeno ključno besedo naši potencialni kupci ali uporabniki naših storitev oz. izdelkov. V ta namen je treba dobro optimizirati spletno stran, saj le tako izboljšamo pozicijo in samo vidnost na spletu ter s tem izboljšamo obiskanost naše spletne strani.

Optimizacija spletne strani je prevod angleškega izraza »Search Engine Optimization«, ki se v tujini pogosto uporablja kar kot kratica SEO. Pomeni namreč proces oblikovanja in spreminjanja spletnih strani z namenom doseči boljšo vidljivosti in s tem obiskanost spletne strani. (Vidmar 2006)

Za uspešno optimizacijo moramo upoštevati naslednja pravila:

- naslov strani
 - definiramo unikatni naslov spletne strani, ki se nahaja v glavi HTML dokumenta;
 - primer: <title>Fov zapiski</title>;
 - vsaka podstran naj ima unikatni naslov;
 - naslov naj se začne s ključno besedo in naj ne presega 60 znakov ali 3–4 besede.

- meta oznake za opis

v glavi HTML dokumenta v znački »meta keywords« vstavimo ključne besede, ki opisujejo spletno stran, dodati je treba alinejo (enako kot zgoraj)

primer: `<meta name="keywords" content="nasvet,optimizacija" />`

v značko »meta description« vstavimo opis naše spletne strani, ki naj ne presega 200 besed.

Primer: `<meta name="description" content="Pomoč pri optimizaciji spletnih strani" />`

- imena slik povezav in dokumentov
 - imena slik in datotek ali dokumentov poimenujemo ali vključimo ključne besede;
 - namesto »img000.jpg« in home.html raje »optimizacija.jpg« in optimizacija.html ipd.

- ime domene
 - na samo pozicijo lahko vplivamo že s ustreznim imenom domene.
Primer: »optimizacija.si«.

- vsebina spletne strani
 - v samo vsebino spletne strani dodamo ključne besede. Besede, ki se pojavijo na začetku, in tiste, ki se večkrat ponovijo, imajo večjo težo od ostalih besed in tako nas iskalniki lažje najdejo. Paziti moramo, da ne presežemo 7 % delež med vsemi besedami v vsebini, saj nas tako iskalniki kaznujejo zaradi spletnega onesnaževanja z izgubo pozicije na spletu.

- povezave z drugih strani
 - po izvršeni vsebinski optimizaciji moramo poskrbeti, da bodo našo spletno stran roboti našli in jo indeksirali ter vključili v svojo podatkovno bazo;
 - spletno stran lahko vpišemo v veliko iskalnikov in imenikov, npr. Google web master tools, najdi.si, ibusinesspromoter.com, submitwolf.net;

- poskrbimo za dodatne izmenjave povezav, ki kažejo na našo stran s pomočjo poslovnih partnerjev;
- uporaba socialnih omrežij;
- pozicija na iskalnikih je odvisna od števila povezav in kvalitete strani, kjer so te povezave objavljene

Vir: (<http://www.nasvet.com/optimizacija-spletnih-strani/>, 24. 11. 2013).

3. SPLNETNI JEZIKI IN ORODJA

3.1. HTML

HTML je označevalni jezik, ki nam predstavlja temeljni kamen in standard pri izdelavi spletnih strani. Sintaksa je zelo preprosta, saj je jezik sestavljen iz značk, ki sestojijo iz lomljenih oklepajev v parih, npr. začetna značka `<p>` in zaključek značke `</p>`. Znotraj te značke po navadi vstavimo tekst. Pri postavitvi značk moramo paziti na pravilno gnezdenje, saj nam v nasprotnem primeru brskalnik vrne napako.

Primer osnovnih gradnikov HTML:

```
<html>
  <head>
</head>
  <body>
    <p>Pozdravljeni.</p>
  </body>
</html>
```

Z začetnim ukazom `html` spletni brskalnik prepozna spletno stran, konča pa se s `html`.

Sledi ukaz glava »*head*« in »*/head*«, ki je namenjen strežniškim programom in je uporabniku na spletni strani neviden. Ukaz telo »*body*« in »*/body*« je tisti del vsebine, ki jo vidimo na spletni strani.

Posamezni znački (elementu) lahko dodajamo tudi attribute, kjer dodajamo lastnosti za prikaz. S tem elementu povemo, kako naj se prikaže na spletni strani. Vir: (<http://mrvar.fdv.uni-lj.si/sola/VIS2/html/htmlslo.htm#Kaj>, 20. 5. 2013).

Primer HTML kode: `<div id="primer" style="color:#000000; font-family:Arial; ">`
`</div>`

3.2. PHP

PHP kratica pomeni »*Hypertext Preprocessor*. Je strežniški programski jezik za izdelavo dinamičnih spletnih strani, ki ga uporabljamo v kombinaciji s HTML in številnimi drugimi jeziki v različnih operacijskih sistemih. Druga okrajšava za PHP je *Personal Home Page* oziroma osebna domača stran.

Oče PHP-ja je spletni programer Rasmus Lerdorf, ki ga je leta 1995 tudi napisal, ter ponudil javnosti kot odprto kodo. PHP je zelo priljubljen med razvijalci spletnih strani, saj je zelo prilagodljiv in se z njim lahko izdelajo zelo kakovostne spletne strani. PHP sam po sebi je neuporaben, saj s samo kodo ne moremo narediti nič, če ne delamo v strežniškem načinu. V povezavi s podatkovnimi zbirkami pa pokaže svojo moč. Za izdelavo spletne strani potrebujemo strežniški program, strežniški modul in razčlenjevalnik kode PHP ter brskalnik. Slika 3 prikazuje uradno stran PHP, kjer so na voljo vse informacije in novice za razvijalce in programerje tega jezika.

Za pisanje kode se uporabljajo različni urejevalniki besedil. Strežnik prepozna kodo z začetno oznako »`<?php`« in končno oznako »`?>`« .

Primer osnovne sintakse:

```

<html>
  <head>
</head>
  <body>
    <?php echo "Pozdravljeni"; ?>
  </body>
</html>

```

Dokument, sestavljen s PHP kodo, shranimo s končnico .php in zaženemo z brskalnikom.

The screenshot shows the official PHP website with a purple header. The main content area features several news items:

- PHP 5.4.16 and PHP 5.3.26 released!**: A news item dated 06-Jun-2013 announcing the availability of PHP 5.4.16 and 5.3.26, which fix about 15 bugs, including CVE-2013-2110. It encourages users to upgrade to PHP 5.4.16 and provides links to source downloads, Windows binaries, and the changelog.
- PHP 5.5 RC3 is available**: A news item dated 06-Jun-2013 announcing the availability of PHP 5.5 RC3, which fixes some bugs against RC2. It includes a warning: **THIS IS A DEVELOPMENT PREVIEW - DO NOT USE IT IN PRODUCTION!** and provides a link to an incomplete changelog.
- PHP 5.5.0RC2 is available**: A news item dated 23-May-2013 announcing the availability of the second release candidate of PHP 5.5, which fixes some bugs against RC1 and improves overall stability. It also includes a warning: **THIS IS A DEVELOPMENT PREVIEW - DO NOT USE IT IN PRODUCTION!**

The right sidebar contains sections for **Stable Releases** (listing 5.4.16 and 5.3.26), **Upcoming Events** (listing various conferences like Spanish Symfony Conference, Venez découvrir PHP 5.5, and Shanghai PHP conference 2013), and **User Group Events** (listing various user group meetings).

Slika 3: Uradna spletna stran za razvijalce PHP-ja

Vir: (<http://www.php.net>, 20. 5. 2013)

3.3. CSS

Kratice CSS pomeni »*Cascading Style Sheet*« ali slovensko kaskadne stilske podloge. CSS je preprost slogovni jezik, ki skrbi za boljšo predstavitev spletne strani. Z njim lahko

določamo barve, velikosti, odmike, poravnave, obrobe, pozicije ter vrsto drugih atributov. Nadziramo lahko aktivnosti, ki jih uporabnik izvaja nad elementi strani (npr. prekritje povezave z miško).

Glavni namen CSS-ja poleg definiranja pravil je predvsem v ločitvi strukture strani, ki jo podaja označevalni jezik skupaj z vsebino njene predstavitve. Množice strani lahko uporabljajo iste podloge, s čimer se izognemo nepotrebemu podvajanju kode ter omogočimo lažje urejanje in dodajanje stilov, poskrbimo pa tudi za večjo preglednost dokumentov, temelječih na HTML »sintaksi«. Rezultat tega je manjša velikost datotek in s tem hitrejša delovanje spletne strani.

Vir: (http://www.egradiva.net/moduli/spletne_aplikacije/css/30_css/04_datoteka.html, 20. 5. 2013).

3.4. MYSQL

Zbirka podatkov je skupek informacij, ki se navezujejo na določene predmete ali namen, kot je recimo beleženje naročil strank ali urejanje glasbene zbirke. (Werber, 2006)

MySQL je odprtokodni sistem za upravljanje z relacijskimi podatkovnimi zbirkami, ki lahko deluje samostojno kot podatkovni strežnik v kombinaciji z drugimi tehnologijami. Deluje po principu odjemalec – strežnik in za delo s podatki uporablja jezik SQL. MySQL je razvilo švedsko podjetje MySQL AB, kot odprtokodno rešitev pod pogoji uporabe javne licence GNU. Vir: (<http://www.gnu.org> 20. 5. 2013). Napisan je v programskem jeziku C in C++.

Po navedbah Štancar in Klemen (2005) so glavne značilnosti MySQL naslednje:

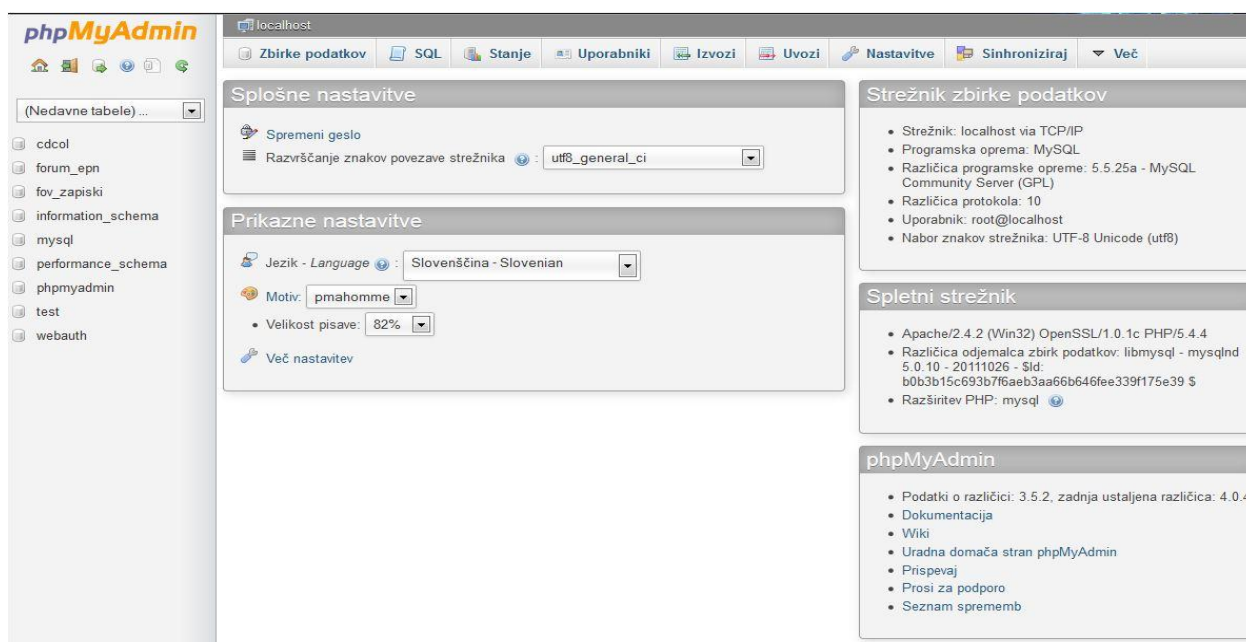
- podpira večprocesorski način delovanja;
- podpira različne programske vmesnike: C, C++, Eiffel, Java, Perl, PHP, Python Intcp;
- podpira različne operacijske sisteme;

- podpira več različnih in daljših nizov v stolpcih;
- hitro združevanje podatkov;
- popolna podpora ukazov Select in Where;
- mešanje tabel iz različnih zbirk pri povpraševanju za podatki;
- šifrirana gesla omogočajo varno upravljanje na daljavo;
- 32 indeksiranih vnosov po 500 bajtov na tabelo;
- fiksna in prilagodljiva dolžina vpisov;
- začasne tabele v pomnilniku;
- velikost zbirke (5 milijard vrstic in več) je dejansko omejena samo z močjo računalnika;
- podpira različne pisave in nabore znakov;
- pomoč je zelo dobro podprta v spletni obliki;
- GB + podatkov na tabelo (odvisno od operacijskega sistema);
- javna licenca GNU ter
- podpira različne nabore znakov.

3.5. PHPMYADMIN

Ker je lahko administracija strežnika MySQL zapleteno opravilo, si preko ukazne vrstice s pomočjo PhpMyAdmin olajšamo delo. Slika 4 prikazuje grafični vmesnik MySQL.

PhpMyAdmin je skupek programskih kod PHP, s katerim dostopamo do strežnika MySQL. Omogoča nam izdelovanje, dodajanje, brisanje in urejanje zbirk podatkov, tabel in podatkov na strežniku MySQL (Štancar, Klemen 2005).



Slika 4: PhpMyAdmin urejevalnik podatkovne zbirke

3.6. XAMPP

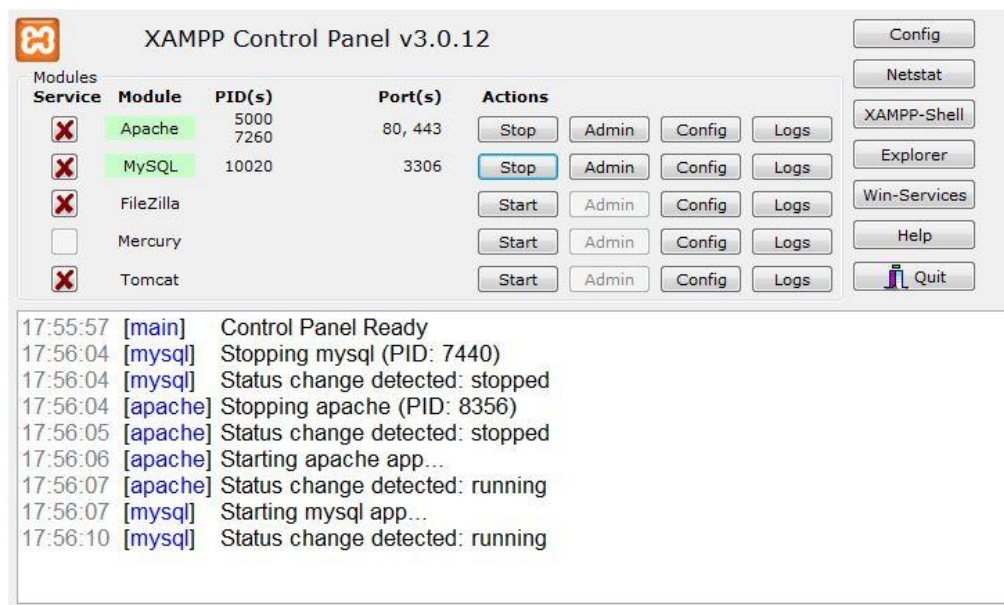
Akronim Xampp je odprtokodna programska oprema, ki teče tudi na operacijskem sistemu Windows, oba skupaj pa tvorita delujoč spletni strežnik za gostovanje dinamične spletne strani na lastnem računalniku.

Paket sestavljajo:

- Windows ali Linux kot operacijski sistem,
- Apache kot spletni strežnik,
- MySQL kot strežnik podatkovne baze,
- Perl, PHP, Python kot programski jeziki.

XAMPP dobimo na spletni strani. Vir: (<http://www.apachefriends.org/en/xampp-windows.html>, 20. 5. 2013),

Slika 5 predstavlja prikaz okna strežnika XAMPP, preko katerega lahko upravljamo strežnik.



Slika 5: Okno strežnika XAMPP

4. STRUKTURA SPLETNE STRANI

V nadaljevanju sledi opis postopka izdelave grafičnega oblikovanja spletne strani, opis izdelave registracije uporabnikov, vmesnika za iskanje ter dodajanje zapiskov v podatkovno bazo.

4.1. NASLOVNA STRAN

Obiskovalci na naslovni strani dobijo prvi vtis o spletni strani, zato mora biti skrbno oblikovana in unikatna. Obiskovalci iščejo različne informacije na spletu, saj želijo rešiti svoj problem. Zato je pomembno, da je na naslovni strani obiskovalcu takoj jasno kakšno korist lahko od le-te pričakuje. V našem primeru so to zapiski. Celotna stran je zelo enostavna in pregledna za uporabo.

Kar zadeva videz naslovne strani, smo se odločili za nežne barve in oglate (zaobljene) oblike elementov. Prav tako smo se izognili prekomerni uporabi teksta. Uporabili pa smo nekaj dinamičnih estetskih elementov, ki stran poživijo.

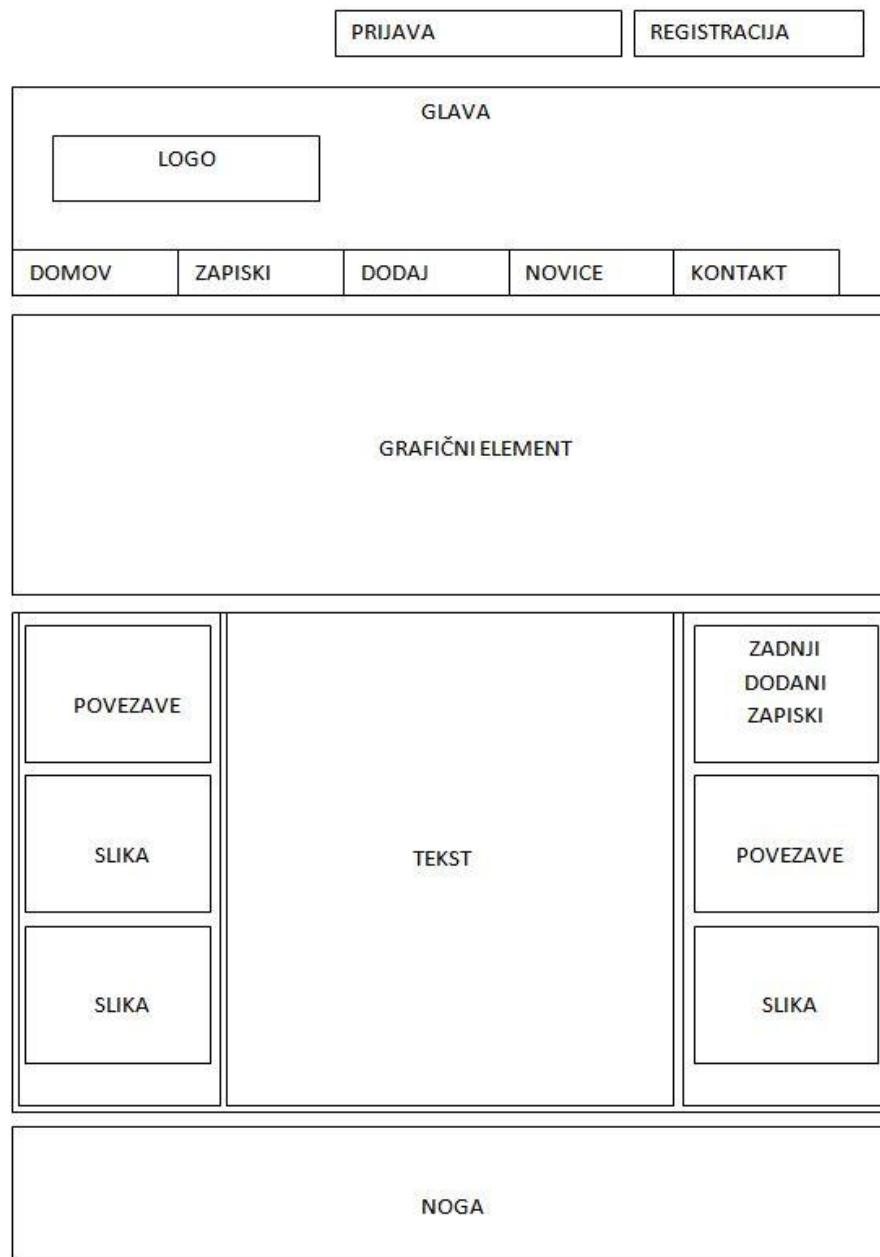
Grafična elementa iz naslovne strani (glava in noga) smo uporabili tudi v drugih podstraneh.

4.1.1. Predloga naslovne strani

Predloga je sestavljena iz klasične postavitve spletne strani. Na vrhu smo dodali glavo spletne strani, ki vsebuje menijske povezave na ostale podstrani. Pod glavo smo dodali prostor za grafični element, ki ima zgolj estetski namen in simbolizira knjigo z zapiski. Osrednji del je namenjen tekstu in različnim povezavam; ta del smo razdelili na tri dele. Prvi levi sklop je namenjen povezavam na različne spletne strani in slikam.

Sredinski del je namenjen tekstu za različna sporočila obiskovalcem. Desni sklop je rezerviran za dinamični prikaz zadnjih dodanih zapiskov, kjer lahko obiskovalci dobijo hitri vpogled v to, ali so dodani novi zapiski. Sledijo še dodatne povezave, ki vodijo obiskovalca na različne podstrani in ostale povezave, namenjene izobraževanju spletnih tehnologij. V spodnjem delu je noga spletne strani kot zaključena celota.

Slika 6 prikazuje predlogo naslovne strani.



Slika 6: Predloga naslovne strani

4.1.2. Grafično oblikovanje

Pri izdelavi spletne strani smo upoštevali dejstvo, da mora biti dizajn kar se da privlačen in hkrati uporabniku omogočati enostavno uporabnost in preglednost. V ta namen smo se odločili izdelati enostavno grafično predlogo za lažjo uporabo spletne strani.

Za ozadje spletne strani smo izdelali dva grafična elementa (glej sliko 7 s programom Photoshop) in jima dodali rahli prehod. V postavitvi na spletno stran smo jima dodali ponavljanje na x osi in se tako izognili dodajanju nepotrebne velikosti slike za celotno ozadje (slika 8). Tako lahko brskalnik hitreje naloži in prikaže spletno stran.

Primer CSS kode za izdelavo ozadja:

```
.body {  
  width:100%;  
  font-family:Arial;  
  height:100%;  
  background:url('slike/body-ozd.png'),url('slike/body-ozd2.png');  
  background-repeat:repeat-x;  
  background-position:top,bottom;  
  margin-top:0px;  
  margin:0 auto;  
  -min-height:1600px;}
```



Slika 7: Grafična elementa za ozadje



Slika 8: Ozadje spletne strani

Naslovna stran vsebuje glavo, kjer je logotip spletne strani, ter dodane barve in grafični element mreža kot estetsko dopolnitev (slika 9).



Slika 9: Glava naslovne strani

S CSS predlogo smo na zgornjem delu slike naredili zaobljene robove, ki dajejo prijeten videz samemu elementu (slika 10).

Primer CSS kode:

```
.zaobljeno {  
    -moz-border-radius-topleft:15px;  
    -moz-border-radius-topright:15px;  
    border-top-left-radius: 15px;
```



```
border-top-right-radius: 15px;}
```



Slika 10: Glava z zaobljenimi robovi

Za menijske povezave, ki vodijo uporabnika na zelene podstrani, smo izdelali gumbе v programu Photoshop. Menijske gumbе smo uporabili tako, da se prekrivajo med seboj z uporabo CSS. To nam omogoča z-indeks, kjer lahko nastavimo položaj gumba.

Ustvarili smo razred, ki ga določimo s piko `.gumb` ter mu dodali lastnosti, kot so širina, višina, ozadje itd.

Položaj gumba smo uredili z določanjem vrednosti z-indeksa. V našem primeru je to prvi gumb, ki ima najvišjo vrednost »z-index: 100;« v osredju, temu sledijo ostali, ki so razvrščeni po padajočem vrstnem redu.

Primer CSS kode:

```
.gumb { width:153px;
        height:36px;
        background:url('slike/gumb.png');
        display:block;line-height:36px;
        text-align:center;float:left;
        position:absolute;
        margin-top:119px;
        color:#ffffff;
        font-weight:bold;
        font-size:14px;
```

```
text-decoration:none;
text-shadow:1px 1px #3557e8; }
.polozaj{z-index:100;}
.polozaj1{z-index:99;margin-left:135px;}
.polozaj2{z-index:98;margin-left:275px;}
.polozaj3{z-index:97;margin-left:410px;}
.polozaj4{z-index:96;margin-left:540px;}
```

S prehodi z miško nad element gumb se nam prikažejo različni gumbi. V osnovni postavitvi je viden gumb1 (glej sliko 11). Ob prehodu z miško nad gumb1 se nam prikaže gumb3 (glej sliko 13). S klikom na gumb1 se nam prikaže gumb2 (slika 12). To nam omogoča CSS predloga, ki ima elementa »hover« in »active«. Z uporabnikovo interakcijo pokličemo razred .gumb, ki ima določene lastnosti, in nam le-te prikaže na ekranu.

Primer kode CSS:

```
.gumb:hover{background:url('slike/gumb3.png');}
.gumb:active{background:url('slike/gumb2.png');}
```



Slika 11: Gumb1



Slika 12: Gumb2



Slika 13: Gumb3

Slika 14 prikazuje končno postavitev menija.



Slika 14: Končna postavitev menija

Uporabili smo JQuery vtičnik, ki prispeva zgolj k estetiki spletne strani; simbolizira knjigo, ki se nanaša na zapiske (slika 15).



Slika 15: JQuery vtičnik

Vir: (http://codecanyon.net/item/flipper/full_screen_preview/2219655 12.6.2013)

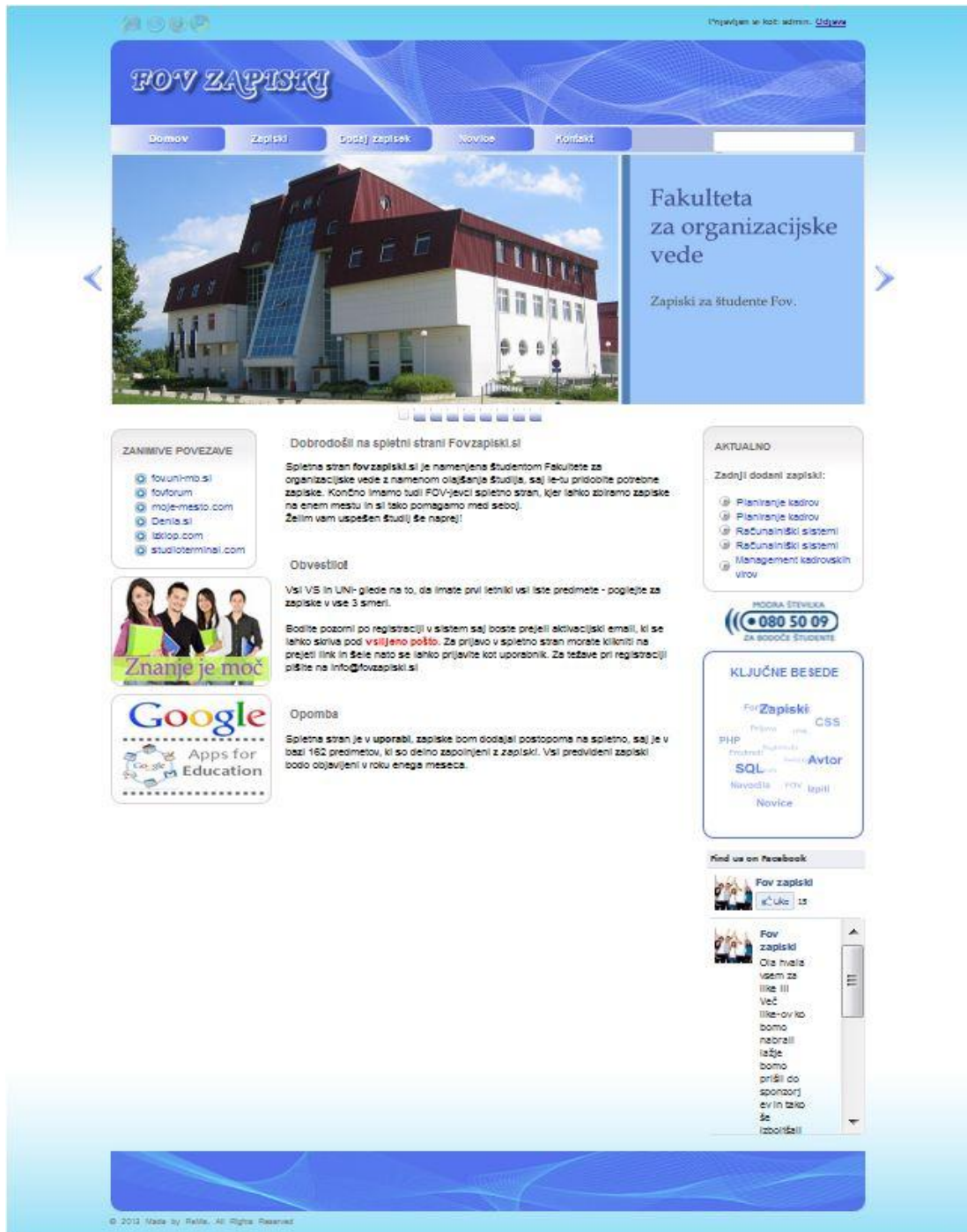
Slika 16 prikazuje vsebinski del naslovne strani, kjer se nahajajo dodatne informacije in obvestila o novostih spletne strani ter razne povezave na različne spletne strani. V stolpcu Aktualno so prikazani zadnji dodani zapiski.

 A screenshot of the main content area of the website. It features a grid of several widgets:

- ZANIMIVE POVEZAVE:** A list of links including fov.uni-mb.si, fovforum, moje-mesto.com, Denia.si, izklop.com, and studioterminal.com.
- Obvestilo!!:** A notice about the university year (VS in UNI) and registration details, accompanied by a photo of a woman and a banner for Gorenjska Banka.
- Opomba:** A note about the website's content and the use of Google Analytics.
- AKTUALNO:** A section titled "Zadnji dodani zapiski:" listing recent articles, all related to "Tehnološki sistemi z osnovami ekologije".
- MODRA ŠTEVILKA:** A contact number 080 50 09 for incoming students.
- KLJUČNE BESEDE:** A cloud of keywords including "Prijava", "SQL", "Zapiski", "Forum", "Novice", "Registracija", "Predmeti", "Avtor", "Kontakt", "HTML", "Izpit", "FOV", "IP", "Navodila", and "Avtor".

Slika 16: Vsebinski del naslovne strani

Na sliki 17 je prikazana celotna naslovna stran.



Slika 17: Videz naslovne strani

4.2. REGISTRACIJA

Da bi obiskovalci lahko dostopali do zapiskov, se morajo registrirati, vendar pa je zaradi varnosti omogočena zaščita pred zlorabo nalaganja napačnih zapiskov, zato je spletna stran namenjena izključno študentom Fakultete za organizacijske vede v Kranju. Narejena je tako, da študente z nepravilno vpisno številko sistem zavrne.

Kadar pa gre za pravilno vpisno številko in izpolnjena vsa vnosna polja, v kateraje za uspešno registracijo treba pravilno vnesti vse zahtevane podatke, nam sistem na pravilno registrirani elektronski naslov pošlje elektronsko sporočilo z aktivacijsko povezavo. Uporabnik s klikom na le-to aktivira svoj uporabniški račun.

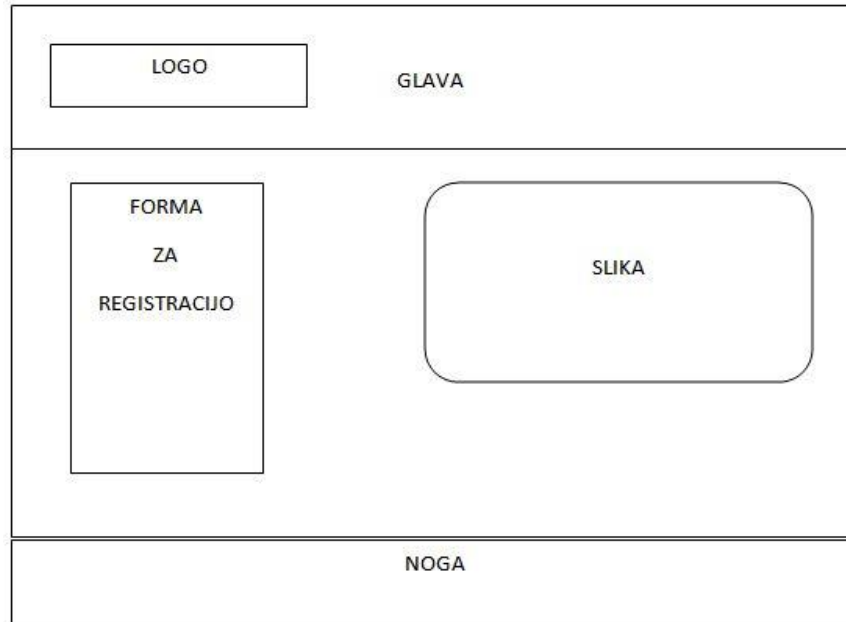
Registracija je sestavljena iz sedmih vnosnih polj, in sicer:

1. imena,
2. priimka,
3. vpisne številke,
4. gesla,
5. ponovitve gesla,
6. elektronskega naslova in
7. ponovitve elektronskega naslova.

Spletna stran je dinamična, zato smo jo naredili tako, da lahko uporabniki samostojno izvedejo proces registracije brez interakcije administratorja. Da ne bi prihajalo do nepravilnosti, smo naredili dodatno zaščito, in sicer ponovitev gesla in elektronskega naslova.

4.2.1. Grafična predloga

Predloga za registracijo je enostavna za izdelavo. Vsebuje glavo in nogo naslovne strani, srednjemu delu smo dodali levo formo za registracijo in desno sliko, ki simbolizira registracijo uporabnikov (glej sliko 18).



Slika 18: Predloga za registracijo

4.2.2. Izdelava registracije

Vnosna polja smo izdelali s pomočjo HTML-ja, ki skupaj tvorijo formo za vnos podatkov.

Primer kode obrazca za formo:

```
<form method='POST' style="margin-left:40px"; font-size:13px; font-family:Vrinda;
action="<?php echo $_SERVER["PHP_SELF"];?> " >
```

Ime:


```
<input type='text' name='ime'><p/>
```

Priimek:


```
<input type='text' name='priimek'><p/>
```

Vpisna številka:


```
<input type='text' name='username'><p/>
```

Geslo:


```
<input type='text' name='password'><p/>
```

Ponovi geslo:


```
<input type='text' name='password2'><p/>
```

Email:


```
<input type='text' name='email'><p/>
```

Ponovi email:


```
<input type='text' name='email2'><p/>
```

```
Input type='submit' name='register' value='Registriraj' >  
</form>
```

Za samo delovanje smo uporabili programski jezik PHP, kjer smo nastavili potek zaščite in pogoje, ki morajo biti izpolnjeni.

Če polja vpisna številka, geslo, elektronski naslov, ime in priimek niso izpolnjena, prejmemo sporočilo: "Izpolnjena morajo biti vsa polja".

Sledi preverba, ali se geslo in elektronski naslov ujemata. S tem zagotovimo, da uporabnik vnese pravilno geslo in elektronski naslov.


Za obiskovalce, ki niso študentje Fakultete za organizacijske vede v Kranju, smo uporabili dodatno varnost vnosa pravilne vpisne številke. Uporabili smo PHP funkcijo "*strlen*", ki nam vrne dolžino znakov. Če vpisna številka ni enaka dolžini 8 znakov, nas sistem zavrne. Prav tako smo uporabili funkcijo »*substr*«, ki ima 3 parametre:

spremenljivko, začetni znak in končni znak. Začetni znak nam pove, od kod naj začne brati znake; končni pa, do kod naj jih bere . Če vpisna številka nima prve 3 številke enake 410, nas sistem zavrne.

Primer kode:

```
if (!$username || !$password || !$email || !$ime || !$priimek)
    $_SESSION['sporocilo'] = "Izpolnjena morajo biti vsa polja!";
elseif($password != $password2)
    {$_SESSION['sporocilo'] = "Gesli se ne ujemata";
    }
elseif($email != $email2)
    {$_SESSION['sporocilo'] = "Emaila se ne ujemata";
    }
elseif(!(strlen($username) == 8 && substr($username, 0, 3) == "410"))
    {$_SESSION['sporocilo'] = "Index ni veljaven";
    }
```

Slika 19 prikazuje končni obrazec za registracijo uporabnikov.



The image shows a web page for user registration. At the top, there is a blue header with the text "FOV ZAPISKI" in a stylized font. Below the header, the word "REGISTRACIJA" is displayed in a smaller font. The main content area contains a registration form with the following fields: "Ime:" (Name), "Priimek:" (Surname), "Vpisna številka:" (Enrollment number), "Geslo:" (Password), "Ponovi geslo:" (Repeat password), "Email:", and "Ponovi email:" (Repeat email). Each field is followed by a white input box. Below the form is a "Registriraj" button. To the right of the form is a 3D illustration of a queue of seven stylized human figures in various colors (red, blue, brown, green, black, yellow, grey) standing behind a metal railing. At the bottom of the form area, there is a blue text box with the message: "Po uspešni registraciji preverite svoj email pod vsiljeno pošto- spam in potrdite link. Brez tega prijava ni možna!". At the very bottom of the page, there is a small copyright notice: "© 2013 Made by ReMa. All Rights Reserved".

Slika 19: Podstran za registracijo uporabnikov

4.3. ISKALNIK ZAPISKOV

Po uspešni prijavi uporabnika s klikom na menijski gumb zapiski povezava vodi na podstran zapiski.php. Stran je namenjena izbiri specifičnega zapiska uporabnika. Za dinamičen prikaz vsebine smo ustvarili relacijsko podatkovno bazo, kjer se izpišejo samo podatki, ki jih določen študent potrebuje.

4.3.1. Opis vmesnika za izbiro zapiskov

Vmesnik za izbiro zapiskov (glej sliko 20) je zelo preprost za uporabo, saj študentu omogoča izbiro svojega programa, smeri, letnika in predmetater poišče želene zapisek. Ob kliku na iskani predmet se nam izpišejo vsi zapiski tega predmeta, ki se nahajajo v podatkovni bazi.

Program:	Smer:	Letnik:	Predmet:
Visokošolski	OMIS	1.let-omis-vs	Statistika
Statistika_VS_Primeri_izpitnih_nalog_.pdf			
StatVSGradivo.pdf			
KMBT36220110526090332.pdf			
izpitna_vprasanja.pdf			
izpiti_2010.pdf			

Slika 20: Vmesnik za izbiro zapiskov

4.3.2. Sistem delovanja vmesnika za izbiro zapiskov

Vmesnik za izbiro zapiskov vsebuje štiri tabele: Program, Smer, Letnik in Predmet. Tabeli Program smo dodali dve polji: visokošolski ter univerzitetni, obe vsebujeta svojo smer študija.

Tabela Smer vsebuje polja: OMIS, OMPDS ter OMKIS. Vsaki od teh pripadajo svoji Letniki. Vsak Letnik ima svoje Predmete in vsak Predmet ima svoje zapiske.

Posebnost, ki jo lahko izpostavimo, je pri izbiri posameznih menijih, kjer se nam izpišejo vsi podatki, ki so v bazi. Za uspešno izbiro zapiska moramo izbrati ustrezen Program, Smer, Letnik in Predmet.

Vmesnik uporablja PHP za polnjenje polj v spustnem meniju. Za prikaz imena uporablja ime programa, smeri, letnika ali predmeta; za vrednost pa se uporabi naslov, ki vsebuje vse dosedanje izbore, ter uporabnikov izbor. Ob spremembi izbora v meniju je uporabnik preusmerjen na naslov, ki je shranjen kot vrednost.

Primer kode vmesnika za izbiro predmeta:

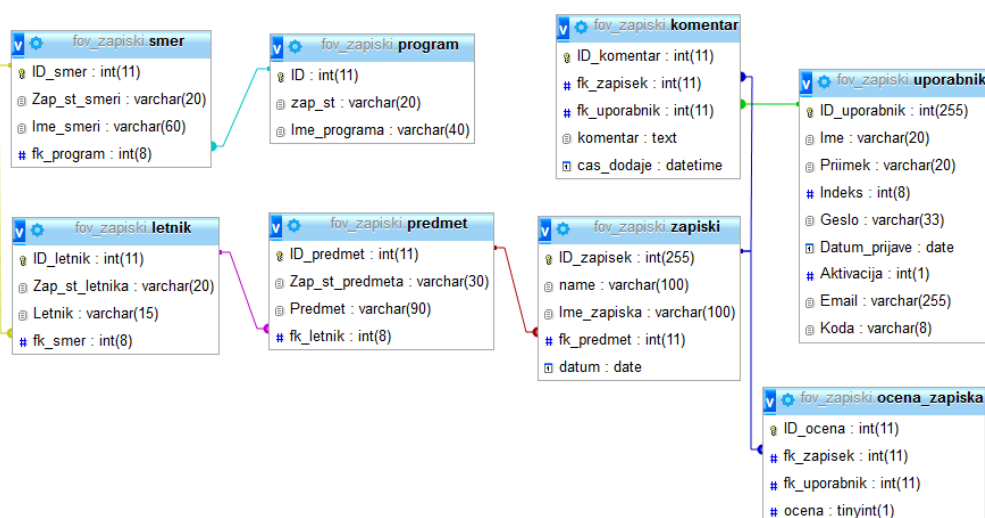
```
<!-- predmet -->
<div class="container">
  <FORM NAME='nav3' action="zapiski.php" method="GET" >
    Predmet:<SELECT style='width:200px;' NAME='SelectURL'
onChange='document.location.href=document.nav.SelectURL.options[document.nav3.Se
lect URL.selectedIndex].value'>
      <option value="zapiski.php?show=<?php echo $_GET['show']?>&show2=<?php echo
$_GET['show2']?>&show3=<?php echo $_GET['show3'] ?>">Izberi predmet</option>
      <?php
mysql_select_db("fov_zapiski", $con);
      $result = mysql_query("SELECT * from predmet where predmet.fk_letnik='
".$show." ' ");
$_ID = null;
      if(isset($_GET['zapisek'])) $_ID= $_GET['zapisek'];
      while($row = mysql_fetch_array($result))
      {
echo "<"<OPTION VALUE='zapiski.php?show=$_GET[show]&show2=$_GET[s
ho
w2]&show3=$_GET[show3]&zapisek=".$row['ID_predmet']."' ";
      if ($_ID===$row['ID_predmet']) echo "selected";
      echo ">".$row['Predmet']."</OPTION>";
      }
      #$_SESSION['$index']=$show;
      ?>
    </SELECT>
  </FORM>
</div>
```

4.3.3. Izdelava podatkovne baze

Spletna strani vsebuje bazo podatkov, ki temelji na tehnologiji MySQL. Je odprtokodna rešitev ter enostavna implementacija z uporabo programskega jezika PHP.

Relacijska podatkovna baza vsebuje 8 tabel (glej sliko 21), ki vsebujejo attribute za pravilno delovanje in funkcionalnost spletne strani. Za izdelavo koncepta smo uporabili šolski sistem, ki ga študent obiskuje.

Šolski program vsebuje visokošolski in univerzitetni program; ta pa se delita na tri smeri, OMIS (Organizacija in management informacijskih sistemov), OMPDS (Organizacija in management poslovnih in delovnih sistemov) in OMKIS (Organizacija in management kadrovskih informacijskih sistemov). Vsaka smer ima tri letnike in vsak letnik ima svoje predmete. Predmeti vsebujejo svoje zapiske. Zapiski vsebujejo komentarje ter ocene, ki jih lahko dodaja uporabnik. Tabela uporabnik je v osnovi izdelana za registracijo uporabnikov.



Slika 21: Relacijska baza podatkov

4.4. PRIKAZ ZAPISKA

Po izbiri predmeta in zapiska, ki ga želimo, nas povezava vodi do samega zapiska, ki ga lahko prenesemo na svoj računalnik. Omogočili smo tudi, da študentje lahko za vsak zapisek dodajo komentar in dobijo informacijo glede zapiska, tj. ali zapisek vsebuje morebitno pripombo ali pohvalo. Dodali smo tudi dva simbola, ki ponazarjata, ali je

posameznemu študentu zapisek všeč ali ne. S klikom na enega se sešteva njihovo skupno število, tako da dobi študent boljše informacijo, ali je zapisek dober ali slab.

4.4.1. Sistem delovanja

Ob poizvedbi zapiska je uporabnik preusmerjen na naslov, ki vsebuje ID zapiska. Ob zahtevi naslova vmesnik uporabi ID zapiska za prenos vnosa, komentarjev ter ocen zapiska iz baze.

Iz vnosa zapiska se uporabi ime datoteke na strežniku za izdelavo naslova za prenos.

S pomočjo ocen se poleg ikone smeškov prikaže (seštevanje) število.

Za komentarje je uporabljena zanka, ki se izvede za vsak komentar, in jih vse tudi izpiše. Vsak komentar vsebuje tudi ID uporabnika ter čas, kdaj je bil komentar dodan.

Za vsak komentar v bazi sistem poišče ime uporabnika, ki ga je dodal.

Stran ima obrazec za dodajanje komentarjev, ki kaže na trenutni naslov. Ta posreduje komentar, ki ga strežnik skupaj z ID zapiska v naslovu in ID uporabnika dobi iz sezone, ter trenutnim časom zapiše v bazo.

Ime zapiska:

[relacije.JPG](#) 😊 1 je všeč 😞 0 ni všeč

Nov komentar: admin

Objavi: admin dne: 10.06.2013 ob 15:37:29

Test dela :)

Slika 22: Vmesnik za dodajanje in izpis komentarjev

Primer kode za komentarje:

```
<form id="komentar" method='POST' style="position:relative;" >
  <p> Nov komentar: <?php echo $_SESSION['ime']; ?> </p>
  <hr>
  <textarea name="Komentar;" ></textarea>
  <input type="submit;" value="Objavi;" style="margin-top:8px;" >
</form>
</div>
<?php
if (isset($_POST['Komentar'])&& trim( $_POST['Komentar'])!=' '){
  $komentar = addslashes(strip_tags($_POST['Komentar']));
  mysql_query("Insert into komentar (fk_zapisek , fk_uporabnik , komentar ,
cas_dodaje) values ($show, {$_SESSION['ID_uporabnik']},{'$komentar}', "" . date("Y-
m-d H:i:s") . "" ) . "" );} ?>
```

```
<table style="color:#888888;">
<?php
$result = mysql_query("SELECT * from komentar where fk_zapisek = '$show'
order by cas_dodaje desc");
while ($komentar = mysql_fetch_array($result))
{
$result2 = mysql_query("SELECT * FROM uporabnik WHERE ID_uporabnik =
" . $komentar['fk_uporabnik']);
$uporabnik = mysql_fetch_assoc($result2);
?>
<tr class="objava"><td>
<div class="objavil" style="color:#4d6eea;"> Objavil: <?php echo
$uporabnik['Ime']; ?> dne: <?php echo date('d.m.Y',strtotime($komentar['cas_dodaje']));
?> ob <?php echo date('H:i:s',strtotime($komentar['cas_dodaje'])); ?></div>
<?php echo str_replace("\n" , "<br>" , $komentar['komentar']); ?>
```

4.5. DODAJANJE ZAPISKOV IN PREDMETOV

Na naslovni strani z menijskim gumbom Dodaj zapisek nas povezava vodi na podstran, kjer lahko vsak posameznik sam dodaja zapise in predmete.

Vmesnik (slika 23) omogoča izbiro programa, smeri, letnika ter predmeta, nato naloži dokument, le-tega pa potrdimo s klikom na gumb Pošlji. Dokument se shrani na strežnik v podatkovno bazo.

Dodaj zapisek:

Program: Izberi program Smer: Izberi smer Letnik: Izberi letnik Predmet: Izberi predmet

Browse... Pošlji Maximalna velikost datoteke je 15MB!

Dodaj predmet:

Program: Izberi program Smer: Izberi smer Letnik: Izberi letnik Ime predmeta: dodaj

Slika 23: Vmesnik za dodajanje zapiskov in predmetov

4.5.1. Sistem delovanja

Ko uporabnik potrdi gumb Pošlji, sistem doda čas, ki je bil dodan, in preveri velikost dokumenta, ki je omejen na 15Mb. Če ni napake, potem se izvedejo funkcije za točni čas, nato pa samemu imenu dokumenta odvzamemo nedovoljene znake in ustvarimo mapo, kjer se dokument začasno shrani. Iz izpolnjenega obrazca razbere predmet in ga z imenom datoteke in datumom doda v bazo zapiskov.

Primer kode za dodajanje zapiska:

```
<!-- vnos zapiska -->
<?php
if (isset($_POST['poslji'])) {
    $datum=date('Y-m-d');
    if($_FILES["file"]["size"]<= 15728640){
        if ($_FILES["file"]["error"] > 0) {
            echo "Prišlo je do napake";
        } else {
            $cas = strtotime("now");
            $ime = preg_replace('/[^\\w\\.]/', "",$_FILES["file"]["name"]);
            mkdir("zapiskpdf/.$cas);
```



```
$doc="zapiskipdf/" . $cas. "/" . $ime;
move_uploaded_file($_FILES["file"]["tmp_name"], $doc);
$deli=explode('=', $_POST['predmet']);
$predmet = addslashes(strip_tags($deli[count($deli)-1]));
$sql="INSERT INTO zapiski (name, Ime_zapiska, fk_predmet,
datum)VALUES(' " . $doc. " ', ' " . $_FILES["file"]["name"]. " ', '$predmet', ' " . $datum. " ' )";
mysql_query($sql);
    }
}
}
?>
```

V spodnjem primeru vidimo prikaz dodanega novega predmeta, pri katerem se doda nov zapis v bazo predmetov. Uporabnik iz obrazca izbere program, smer, letnik in nato v polje vpiše ime predmeta in ga doda v podatkovno bazo.

```
<!-- vnos predmeta -->
<?php
if (isset($_POST['dodaj'])) {
    $deli=explode('=', $_POST['letnik']);
    $letnik = addslashes(strip_tags($deli[count($deli)-1]));
    $predmet = addslashes(strip_tags($_POST['nov_predmet']));
    $sql="INSERT INTO predmet (ID_predmet, Zap_st_predmeta, Predmet,
fk_letnik) VALUES(' ', '$predmet', '$letnik')";

if (!mysql_query($sql,$con))
{
    die('Prišlo je do neznane napake: ' . mysql_error());
}
}
?>
```

5. OPTIMIZACIJA SPLETNE STRANI

Po končani fazi izdelave spletne strani je ključnega pomena ureditev spletne strani, da se prikaže čim višje med rezultati na spletnih brskalnikih. V našem primeru je to brskalnik Google. SEO »Search Engine Optimization« je postopek izboljšanja vidljivosti na spletu ob iskanih rezultatih.

Za povečanje vidljivosti na spletu potrebujemo pravilno izbiro ključne besede, ki najbolje opredeljuje našo dejavnost. V našem primeru smo izbrali besedo »fovzapiski« (glej sliko 23).

Po opredelitvi ključne besede smo se lotili tehnične optimizacije, saj nam ta omogoča dobro pozicijo na spletu. Ključni kriteriji, ki smo se jih lotili so:

- ime domene:

Za ime domene smo izbrali ključno besedo, ki predstavlja unikatni zapis, z le-to pa bistveno izboljšajo pogoji za uvrstitev na spletu.

Domena: www.fovzapiski.si.

- naslov strani:

Naslov strani se nahaja v glavi html dokumenta, v našem primeru:

```
<title>Fov Zapiski</title>
```

Za vse ostale podnaslove velja, da naj bodo unikatni in naj se začnejo s ključno besedo.

- meta podatki:

V glavi HTML dokumenta smo definirali meta podatke »meta keywords, description«, ki se izpišejo pod naslovom strani med izpisanimi rezultati iskanj v brskalniku.

Primer vnosa meta podatkov:

```
<meta name="keywords" content="fovzapiski,fov zapiski,fov, FOV,  
zapisek,študij, snov,Fakulteta, zadnji zapiski, Fakulteta za organizacijske vede,Zapiski,  
Kranj, Študentski svet " />
```

```
<meta name="description" content="Fovzapiski.si je nova spletna stran za študente  
FOV-ja, kjer lahko dobite zapiske ali pa jih dodajate. " />
```

- vsebina spletne strani:

Vsebina spletne strani naj bo ljudem koristna, unikatna, izvirna in kakovostno napisana. Pomembna je avtentična vsebina, ki jo objaviš prvi. Zato tudi uporabili ključne besede v besedilu, ki se jih poudari s HTML značkami za okrepljeno besedilo, na primer: fovzapiski, pa tudi označeni naslovi <h3>Dobrodošli na spletni strani Fovzapiski.si</h3>.

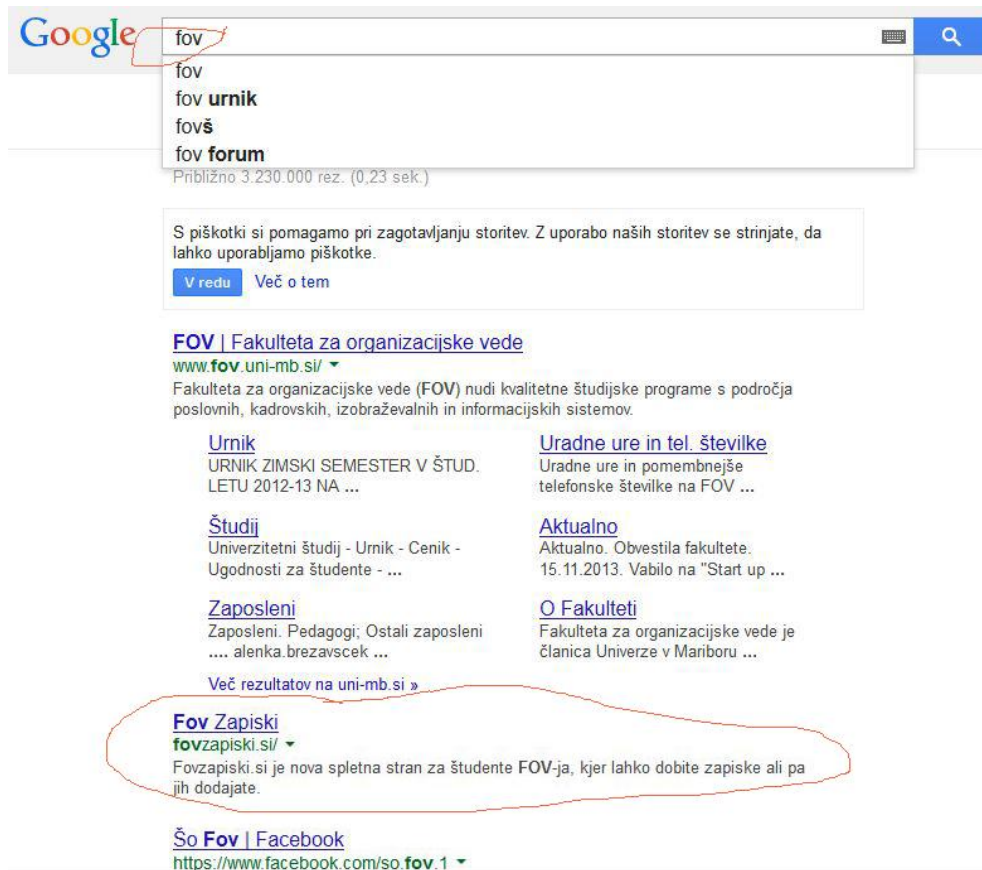
- povezave z drugih strani:

Pomemben faktor pri iskalnih robotih so tudi zunanje povezave, ki kažejo na nas. Za povezave z drugih strani smo si pomagali z izmenjavo spletnih naslovov s Fakulteto za organizacijske vede v Kranju ter prijateljskimi povezavami, kjer naša stran kaže nanje in obratno. Domeno smo vpisali tudi v brskalnik Najdi.si.

- socialna omrežja:

Naredili smo tudi Facebook profil in komponento vgradili na naslovni strani za »všečkanje«. Na podstran Kontakt smo dodali dva Youtube video posnetka, ki se nanašata na temo študentov FOV. Tudi to bistveno pripomore pri sami razvrstitvi rezultatov v brskalnikih.

Rezultat optimizacije:



Slika 24: Primer uvrstitve zadetkov ob vnosu ključne besede

6. ANALIZA SWOT

Kratice SWOT je sestavljena iz začetnic angleških besed:

S – strenghts (prednosti),

W – weaknesses (slabosti),

O – opportunities (priložnosti),

T – treats (nevarnosti).

Analiza SWOT nam omogoča, da enostavno analiziramo obstoječe stanje v okolju, v katerem se nahajamo, in tako opredelimo svoje prednosti, slabosti, priložnosti in nevarnosti ter dobimo jasno in natančno sliko bodisi podjetja, aplikacije ali dejavnosti v primerjavi s konkurenco. SWOT analiza je široko uporabna, saj jo lahko uporabimo v katerikoli dejavnosti in. Končna strategija je, da gradimo na prednostih, odpravimo pomanjkljivosti, izkoristimo priložnosti ter se izognemo nevarnostim.

Vir: (<http://www.blazkos.com/swot-analiza.php>, 24. 11. 2013).

SWOT analiza spletne strani:

Prednosti:

- naša največja prednost je, da smo vse zapiske združili na eno spletno stran;
- hiter in enostaven ter pregleden vmesnik za iskanje zapiskov;
- dostopajo lahko samo študentje FOV, s čimer minimiziramo neuporabne zapiske;
- dodatna varnostna zaščita.

Slabosti:

- ni foruma, kjer bi se študentje lahko izmenjevali informacije o zapiskih;
- ni preverjanja avtentičnosti dodanega zapiska;
- omejitev sistema na eno univerzo.

Priložnosti:

- razširitev portala za celotno Univerzo v Mariboru;
- uvedba foruma.

Nevarnosti:

- zloraba dodajanja neverodostojnih zapiskov;
- poseganje v sistem.

7. ZAKLJUČEK

V diplomskem delu smo predstavili izdelavo dinamične spletne strani po posameznih korakih, opisali postopke registracije in uporabo vmesnika za dostop do zapiskov ter dodajanje le-teh. Opisali smo postopek optimizacije spletne strani in naredili SWOT analizo.

Glavna ovira, na katero smo naleteli, je, kako dinamično pridobiti zapiske iz baze in ob tem ohraniti pretekle izbore ter zahteve iz baze, ki se med seboj povezujejo. Rešitev smo našli v sestavi naslovov, ki se med seboj združujejo.

Ko smo spletno stran dokončali, smo jo dodali na spletni strežnik društva Denia, kjer smo imeli urejeno spletno gostovanje. Domeno smo zakupili pri spletnem ponudniku Domovanje.com. V bazi se je nahajalo 584 različnih zapiskov. Spletna stran je bila študentom dostopna od aprila 2013, do decembra 2013 pa je imela registriranih že 410 uporabnikov. Kljub temu da je uporabnikom bila dana možnost komentiranja posameznih zapiskov, se tega niso posluževali. Na spletno stran smo implementirali orodje Google Analytics, ki nam je med drugim prikazal tudi promet obiska na spletni strani za časovno določen interval. Do sedaj smo imeli že 32657 ogledov strani.

8. UGOTOVITVE

Spletna stran je učinkovito sredstvo za pomoč pri samem študiju, saj si lahko študentje bistveno povečajo obseg znanja ter učne rezultate. Na spletni strani dobijo bazo znanja ne le za obvezne predmete, temveč tudi za vse ostale, ki so na voljo.

Cilj naloge je dosežen, saj smo naredili dinamično spletno stran, ki komunicira z uporabnikom, in mu postreže podatke, ki jih le-ta potrebuje.

9. MOŽNOSTI NADALJNJEGA RAZVOJA

Spletna stran je dinamična v kombinaciji s podatkovno bazo, kar nam dopušča relativno enostavno nadgradnjo. Trenutno se spletna stran ne razvija več, imamo pa namen v prihodnje povezati vse fakultete Univerze v Mariboru in tako še povečati bazo znanja ter uporabnost spletne strani.

LITERATURA IN VIRI

Knjige:

Kaltenekar, Matic. (2006): Oblikovanje spletnih strani: HTML, CSS in JavaScript, Pasadena, Ljubljana.

Štancar, Matjaž. Klemen, Simon. (2005): PHP in MySQL na spletnem strežniku Apache (2. dopolnjena izdaja), Pasadena, Ljubljana.

Vidmar, D. (2006): Optimizacija spletnih strani, Fakulteta za organizacijske vede, Kranj.

Zandstra, Matt. (2004): Naučite se PHP v 24 urah, Pasadena, Ljubljana.

Werber, Borut (2006): Uporaba MS ACCESS-a 2003 v praksi: skripta, Kranj, Moderna organizacija.

Poročila, interni dokumenti:

Werber, Borut. (2012): Zapiski s predavanj: Metode in tehnike sistemske analize.

Zupančič, Jože. (2007): Prosojnice iz predavanj: Metode in tehnike sistemske analize

Spletne strani:

Dinamična spletna stran: <http://www.mimovrste.com> (15. 9. 2013).

GNU/GPL: <http://www.gnu.org> (20. 5. 2013).

Izdelava dinamičnih spletnih strani:

<http://www.izdelava-spletnih-strani-1a.blogspot.com/2010/06/staticne-ali-dinamicne-spletne-strani.html> (20. 6. 2013).

Izdelava statičnih spletnih strani:

<http://www.izdelava-spletnih-strani-1a.blogspot.com/2010/06/staticne-ali-dinamicne-spletne-strani.html> (15. 8. 2013).

Jquery vtičnik: http://codecanyon.net/item/flipper/full_screen_preview/2219655 (12.6.2013)

Kaj je CSS:

http://www.egradiva.net/moduli/spletne_aplikacije/css/30_css/04_datoteka.html (20. 5. 2013).

Optimizacija spletnih strani: <http://www.nasvet.com/optimizacija-spletnih-strani/> (24. 11. 2013).

Osnove HTML: <http://mrvar.fdv.uni-lj.si/sola/VIS2/html/htmlslo.htm#Kaj> (20. 5. 2013)

Prototipna metoda:

<http://www.swe.uni-linz.ac.at/publications/pdf/tr-se-94.02.pdf> (26. 11. 2013).

Statična spletna stran: <http://www.cisto-je.si> (15. 8. 2013).

SWOT analiza: <http://www.blazkos.com/swot-analiza.php> (24. 11. 2013).

Uradna stran PHP: <http://www.php.net> (20. 5. 2013).

XAMPP za Windows: <http://www.apachefriends.org/en/xampp-windows.html> (20. 5. 2013).

KAZALO SLIK

Slika 1: Primer statične spletne strani	5
Slika 2: Primer dinamične spletne strani	6
Slika 3: Uradna spletna stran za razvijalce PHP-ja.....	11
Slika 4: PhpMyAdmin urejevalnik podatkovne zbirke.....	14
Slika 5: Okno strežnika XAMPP	15
Slika 6: Predloga naslovne strani	17
Slika 7: Grafična elementa za ozadje.....	18
Slika 8: Ozadje spletne strani.....	19
Slika 9: Glava naslovne strani	19
Slika 10: Glava z zaobljenimi robovi	20
Slika 11: Gumb1	21
Slika 12: Gumb2	21
Slika 13: Gumb3	21
Slika 14: Končna postavitev menija	21
Slika 15: JQuery vtičnik	22
Slika 16: Vsebinski del naslovne strani	22
Slika 17: Videz naslovne strani	23
Slika 18: Predloga za registracijo	25
Slika 19: Podstran za registracijo uporabnikov	28
Slika 20: Vmesnik za izbiro zapiskov.....	29
Slika 21: Relacijska baza podatkov	31
Slika 22: Vmesnik za dodajanje in izpis komentarjev	33
Slika 23: Vmesnik za dodajanje zapiskov in predmetov	35
Slika 24: Primer uvrstitve zadetkov ob vnosu ključne besede.....	39

POJMOVNIK

optimizacija (računalništvo): proces izpolnjevanja nekega sistema z namenom večje učinkovitosti in manjše porabe resursov

indeksiranje (spletni iskalnik): izdelovanje seznama datotek ali dokumentov za namene hitrejšega iskanja relevantnih dokumentov za določeno povpraševanje

sintaksa (računalništvo): beseda se uporablja v računalniški znanosti na področju programiranja, kjer pravilni vrstni red in izbrane kombinacije besed, simbolov, elementov ter njihovih parametrov določajo dobro oblikovani izrazi. Sintakso formalnega jezika običajno določa gramatika.

Google Analytics (računalništvo): je analitično orodje za natančen promet spletne strani.

KRATICE IN AKRONIMI

CMS	Content Management System: sistem za upravljanje vsebin
PHP	Hypertext Preprocessor: strežniški programski jezik
HTML	Hyper Text Markup Language: jezik za označevanje nadbesedila
CSS	Cascade Style Sheets: kaskadne stilske datoteke
MySQL	My Structured Query Language: Sistem za upravljanje s podatkovnimi bazami
SEO	Search Engine Optimization: optimizacija za spletne iskalnike
SQL	Standard Query Language: standardni povpraševalni jezik
GNU/GPL	General Public Licence: je licenca računalniškega operacijskega sistema, sestavljena izključno iz prostega programja.